APPENDIX D

IP Addressing Practice

Chapter 4, "IP Addressing," covers many details related to analyzing IP addresses, subnets, and summarized IP routes. That chapter suggests some decimal math algorithms that allow you to find the answers to some typical questions without having to perform timD-consuming conversions between binary and decimal.

As promised in Chapter 4, this appendix provides some practice problems that should help you perfect the use of the algorithms in Chapter 4. Note that the goal of this practice is not to make you memorize the algorithms—instead, the goal is to help you become so familiar with the patterns in the decimal math that you can look at a problem and visualize the answer quickly. The intent is to enable you, after you have practiced enough, to simply look at a problem and do the math in your head, ignoring the specific steps in the book.

This appendix covers the decimal math processes to answer the following four types of questions:

1. Given an IP address and mask/prefix length, list the number of subnets (assuming SLSM), number of hosts per subnet (assuming SLSM), the subnet number, the broadcast address, and the range of valid IP addresses in that same subnet.
2. Given an IP network and a static mask/prefix length, list the subnet numbers.
3. Given a set of routes, find the smallest inclusive summary route.
4. Given a set of routes, find the smallest exclusive summary route(s).

These topics are covered in order in this appendix. Also, check http://www.get-cisco-certified.com, a website at which we intend to offer additional exam preparation help—we will be adding additional practice problems on the web site as time permits.

Subnetting Practice

This appendix lists 25 separate questions, asking you to derive the subnet number, broadcast address, and range of valid IP addresses. In the solutions, the binary math is shown, as is the process that avoids binary math using the "subnet chart" described in Chapter 4, "IP Addressing." You might want to review Chapter 4's section on IP addressing before trying to answer these questions.

25 Subnetting Questions

Given each IP address and mask, supply the following information for each of these 25 examples:

- Size of the network part of the address
- Size of the subnet part of the address
- Size of the host part of the address
- The number of hosts per subnet
- The number of subnets in this network
- The subnet number
- The broadcast address
- The range of valid IP addresses in this network:

1. 10.180.10.18, mask 255.192.0.0
2. 10.200.10.18, mask 255.224.0.0
3. 10.100.18.18, mask 255.240.0.0
4. 10.100.18.18, mask 255.248.0.0
5. 10.150.200.200, mask 255.252.0.0
6. 10.150.200.200, mask 255.254.0.0
7. 10.220.100.18, mask 255.255.0.0
8. 10.220.100.18, mask 255.255.128.0
9. 172.31.100.100, mask 255.255.192.0
10. 172.31.100.100, mask 255.255.224.0
11. 172.31.200.10, mask 255.255.240.0
12. 172.31.200.10, mask 255.255.248.0
13. 172.31.50.50, mask 255.255.252.0

14. 172.31.50.50, mask 255.255.254.0
15. 172.31.140.14, mask 255.255.255.0
16. 172.31.140.14, mask 255.255.255.128
17. 192.168.15.150, mask 255.255.255.192
18. 192.168.15.150, mask 255.255.255.224
19. 192.168.100.100, mask 255.255.255.240
20. 192.168.100.100, mask 255.255.255.248
21. 192.168.15.230, mask 255.255.255.252
22. 10.1.1.1, mask 255.248.0.0
23. 172.16.1.200, mask 255.255.240.0
24. 172.16.0.200, mask 255.255.255.192
25. 10.1.1.1, mask 255.0.0.0

Suggestions on How to Attack the Problem

If you are ready to go ahead and start answering the questions, go ahead! If you want more explanation of how to attack such questions, refer back to the section on IP subnetting in Chapter 4. However, if you have already read Chapter 4, a reminder of the steps in the process to answer these questions, with a little binary math, is repeated here:

NOTE The examples shown here assume classful IP addressing, so the number of subnets per IP network is listed as 2^n - 2. If using classless IP addressing, the numbers would simply be $2^{n.}$

Step 1 Identify the structure of the IP address.

a. Identify the size of the network part of the address, based on Class A, B, and C rules.

b. Identify the size of the host part of the address, based on the number of binary 0s in the mask. If the mask is "tricky," use the chart of typical mask values to convert the mask to binary more quickly.

c. The size of the subnet part is what's "left over"; mathematically, it is 32 – (network + host)

d. Declare the number of subnets, which is $2^{\text{number-of-subnet-bits}} - 2$.

e. Declare the number of hosts per subnet, which is $2^{\text{number-of-host-bits}} - 2$

Step 2 Create the subnet chart that will be used in steps 3 and 4.

a. Create a generic subnet chart.

b. Write down the decimal IP address and subnet mask in the first two rows of the chart.

c. If an easy mask is used, draw a vertical line between the 255s and the 0s in the mask, from top to bottom of the chart. If a hard mask is used, draw a box around the interesting octet.

d. Copy the address octets to the left of the line or the box into the final four rows of the chart.

Step 3 Derive the subnet number and the first valid IP address.

a. On the line on the chart where you are writing down the subnet number, write down 0s in the octets to the right of the line or the box.

b. If the mask is difficult, so that there is a box in the chart, use the magic number trick to find the decimal value of the subnet's interesting octet, and write it down. Remember, the magic number is found by subtracting the interesting (non-0 or 255) mask value from 256. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

c. To derive the first valid IP address, copy the first three octets of the subnet number, and add 1 to the fourth octet of the subnet number.

Step 4 Derive the broadcast address and the last valid IP address for this subnet.

a. Write down 255s in the broadcast address octets to the right of the line or the box.

b. If the mask is difficult, so that there is a box in the chart, use the magic number trick to find the value of the broadcast address's interesting octet. In this case, you add the subnet number's interesting octet value to the magic number, and subtract 1.

c. To derive the last valid IP address, copy the first three octets of the broadcast address and subtract 1 from the fourth octet of the broadcast address.

Question 1: Answer

The answers begin with the analysis of the three parts of the address, the number of hosts per subnet, and the number of subnets of this network using the stated mask. The binary math for subnet and broadcast address calculation follows. The answer finishes with the easier mental calculations using the subnet chart described in Chapter 4.

Table D-1 *Question 1: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Item	Example	Rules to Remember
Address	10.180.10.18	N/A
Mask	255.192.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	22	Always defined as number of binary 0s in mask

Table D-1 *Question 1: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts (Continued)*

Item	Example	Rules to Remember
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 - 2 = 2$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{22} - 2 = 4{,}194{,}302$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-2. To calculate the two numbers, perform a Boolean AND on the address and mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-2 *Question 1: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.180.10.18	0000 1010 10**11 0100 0000 1010 0001 0010**
Mask	255.192.0.0	1111 1111 11**00 0000 0000 0000 0000 0000**
AND result (subnet number)	10.128.0.0	0000 1010 10**00 0000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.191.255.255	0000 1010 10**11 1111 1111 1111 1111 1111**

To get the first valid IP address, just add 1 to the subnet number; to get the last valid IP address, just subtract 1 from the broadcast address. In this case:

10.128.0.1 through 10.191.255.254

10.128.0.0 + 1= 10.128.0.1

10.191.255.255 – 1= 10.191.255.254

Steps 2, 3, and 4 in the process use a table like Table D-3, which lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Figure D-1 at the end of this problem shows the fields in Table D-3 that are filled in at each step in the process. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-3 *Question 1: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Address	10	180	10	18	N/A
Mask	255	192	0	0	N/A
Subnet number	10	128	0	0	Magic number = 256 – 192 = 64

continues

Table D-3 *Question 1: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart (Continued)*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
First address	10	128	0	1	Add 1 to last octet of subnet
Broadcast	10	191	255	255	128 + 64 – 1 = 191
Last address	10	191	255	254	Subtract 1 from last octet

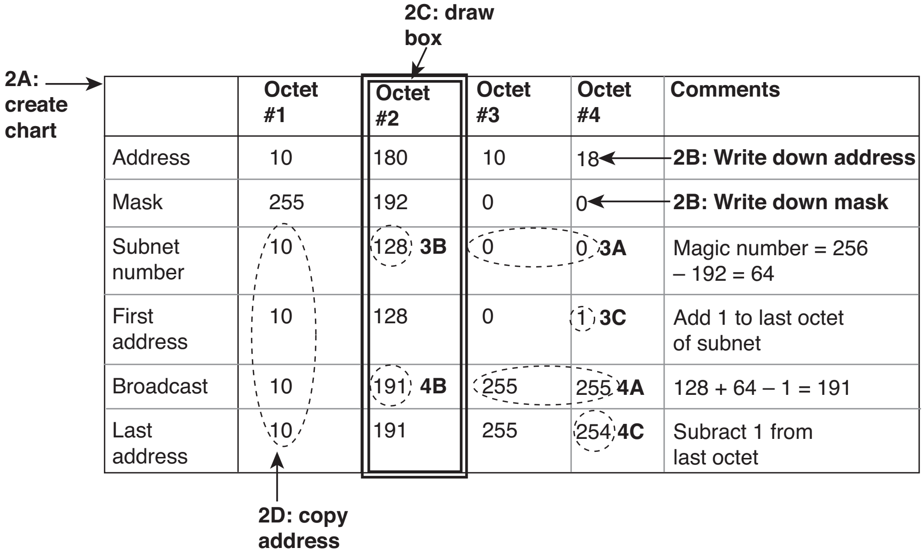
Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet
Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 64 that's closest to 180 but not bigger than 180. So, the second octet of the subnet number is 128.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 128 + 64 – 1 = 191.

Finally, Figure D-1 shows Table D-3 with comments about when each part of the table was filled in, based on the steps in the process at the beginning of the chapter.

Figure D-1 *Steps 2, 3, and 4 for Question 1*



Question 2: Answer

Table D-4 *Question 2: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.200.10.18	N/A
Mask	255.224.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	21	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 - 2 = 6$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{21} - 2 = 2{,}097{,}150$	$2^{\text{number-of-host-bits}} - 2$

Table D-5 presents the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-5 *Question 2: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.200.10.18	0000 1010 110**0 1000 0000 1010 0001 0010**
Mask	255.224.0.0	1111 1111 111**0 0000 0000 0000 0000 0000**
AND result (subnet number)	10.192.0.0	0000 1010 110**0 0000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.223.255.255	0000 1010 110**1 1111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.192.0.1 through 10.223.255.254

Table D-6 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate

the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 32 that's closest to 200 but not bigger than 200. So, the second octet of the subnet number is 192.

Table D-6 *Question 2: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Address	10	200	10	18	N/A
Mask	255	224	0	0	N/A
Subnet number	10	192	0	0	Magic number = 256 – 224 = 32
First address	10	192	0	1	Add 1 to last octet of subnet
Broadcast	10	223	255	255	192 + 32 – 1 = 223
Last address	10	223	255	254	Subtract 1 from last octet

Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet Broadcast rule: Subnet + magic – 1

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 192 + 32 – 1 = 223.

Question 3: Answer

Table D-7 *Question 3: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.100.18.18	N/A
Mask	255.240.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	20	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 - 2 = 14$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{20} - 2 = 1,048,574$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-8. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-8 *Question 3: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.100.18.18	0000 1010 0110 **0100 0001 00100001 0010**
Mask	255.240.0.0	1111 1111 1111 **0000 0000 0000 0000 0000**
AND result (subnet number)	10.96.0.0	0000 1010 0110 **0000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.111.255.255	0000 1010 0110 **1111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.96.0.1 through 10.111.255.254

Table D-9 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-9 *Question 3: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Address	10	100	18	18	N/A
Mask	255	240	0	0	N/A
Subnet number	10	96	0	0	Magic number = 256 – 240 = 16
First address	10	96	0	1	Add 1 to last octet of subnet
Broadcast	10	111	255	255	96 + 16 – 1 = 111
Last address	10	111	255	254	Subtract 1 from last octet

Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate

the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 16 that's closest to 100 but not bigger than 100. So, the second octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 16 – 1 = 111.

Question 4: Answer

Table D-10 *Question 4: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.100.18.18	N/A
Mask	255.248.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	19	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 - 2 = 30$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{19} - 2 = 524{,}286$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-11. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-11 *Question 4: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.100.18.18	0000 1010 0110 0**100 0001 00100001 0010**
Mask	255.248.0.0	1111 1111 1111 1**000 0000 0000 0000 0000**
AND result (subnet number)	10.96.0.0	0000 1010 0110 0**000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.103.255.255	0000 1010 0110 0**111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.96.0.1 through 10.103.255.254

Table D-12 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-12 *Question 4: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4	Comments
Address	10	100	18	18	N/A
Mask	255	248	0	0	N/A
Subnet number	10	96	0	0	Magic number = 256 – 248 = 8
First address	10	96	0	1	Add 1 to last octet of subnet
Broadcast	10	103	255	255	96 + 8 – 1 = 103
Last address	10	103	255	254	Subtract 1 from last octet

Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet
Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 8 that's closest to 100 but not bigger than 100. So, the second octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 8 – 1 = 103.

Question 5: Answer

Table D-13 *Question 5: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.150.200.200	N/A
Mask	255.252.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	18	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 - 2 = 62$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{18} - 2 = 262{,}142$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-14. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-14 *Question 5: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.150.200.200	0000 1010 1001 01**10 1100 1000 1100 1000**
Mask	255.252.0.0	1111 1111 1111 11**00 0000 0000 0000 0000**
AND result (subnet number)	10.148.0.0	0000 1010 0110 01**00 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.151.255.255	0000 1010 0110 01**11 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.148.0.1 through 10.151.255.254

Table D-15 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-15 *Question 5: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	**Octet 2**	**Octet 3**	**Octet 4**	**Comments**
Address	10	150	200	200	N/A
Mask	255	252	0	0	N/A
Subnet number	10	148	0	0	Magic number = 256 – 252 = 4
First address	10	148	0	1	Add 1 to last octet of subnet
Broadcast	10	151	255	255	148 + 4 – 1 = 151
Last address	10	151	255	254	Subtract 1 from last octet

Subnet rule: Multiple of magic number closest to, but not more than, IP address value in interesting octet
Broadcast rule: Subnet + magic – 1

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 148 is the multiple of 4 that's closest to 150 but not bigger than 150. So, the second octet of the subnet number is 148.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 148 + 4 – 1 = 151.

Question 6: Answer

Table D-16 *Question 6: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	**Example**	**Rules to Remember**
Address	10.150.200.200	N/A
Mask	255.254.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C

continues

Table D-16 *Question 6: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts (Continued)*

Step	**Example**	**Rules to Remember**
Number of host bits	17	Always defined as number of binary 0s in mask
Number of subnet bits	7	32 – (network size + host size)
Number of subnets	$2^7 - 2 = 126$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{17} - 2 = 131,070$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-17. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-17 *Question 6: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.150.200.200	0000 1010 1001 011**0 1100 1000 1100 1000**
Mask	255.254.0.0	1111 1111 1111 111**0 0000 0000 0000 0000**
AND result (subnet number)	10.150.0.0	0000 1010 0110 011**0 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.151.255.255	0000 1010 0110 011**1 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.150.0.1 through 10.151.255.254

Table D-18 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-18 *Question 6: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	**Octet 2**	**Octet 3**	**Octet 4**
Address	10	150	200	200
Mask	255	254	0	0

Table D-18 *Question 6: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart (Continued)*

	Octet 1	Octet 2	Octet 3	Octet 4
Subnet number	10	150	0	0
First valid address	10	150	0	1
Broadcast	10	151	255	255
Last valid address	10	151	255	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 254 = 2 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 150 is the multiple of 2 that's closest to 150 but not bigger than 150. So, the second octet of the subnet number is 150.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 150 + 2 – 1 = 151.

Question 7: Answer

Table D-19 *Question 7: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.220.100.18	N/A
Mask	255.255.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	16	Always defined as number of binary 0s in mask
Number of subnet bits	8	32 – (network size + host size)
Number of subnets	$2^8 - 2 = 254$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{16} - 2 = 65{,}534$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-20. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-20 *Question 7: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.220.100.18	0000 1010 1101 1100 **0110 0100 0001 0010**
Mask	255.255.0.0	1111 1111 1111 1111 **0000 0000 0000 0000**
AND result (subnet number)	10.220.0.0	0000 1010 1101 1100 **0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.220.255.255	0000 1010 1101 1100 **1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.220.0.1 through 10.220.255.254

Table D-21 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4.

Table D-21 *Question 7: Subnet, Broadcast, First, and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	220	100	18
Mask	255	255	0	0
Subnet number	10	220	0	0
First valid address	10	220	0	1
Broadcast	10	220	255	255
Last valid address	10	220	255	254

This subnetting scheme uses an easy mask because all of the octets are a 0 or a 255. No math tricks are needed at all!

Question 8: Answer

Table D-22 *Question 8: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.220.100.18	N/A
Mask	255.255.128.0	N/A
Number of network bits	8	Always defined by Class A, B, C

Table D-22 *Question 8: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts (Continued)*

Step	Example	Rules to Remember
Number of host bits	15	Always defined as number of binary 0s in mask
Number of subnet bits	9	32 – (network size + host size)
Number of subnets	$2^9 - 2 = 510$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{15} - 2 = 32{,}766$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-23. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-23 *Question 8: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.220.100.18	0000 1010 1101 1100 0**110 0100 0001 0010**
Mask	255.255.128.0	1111 1111 1111 1111 1**000 0000 0000 0000**
AND result (subnet number)	10.220.0.0	0000 1010 1101 1100 0**000 0000 0000 0000**
Change host to 1s (broadcast address)	10.220.127.255	0000 1010 1101 1100 0**111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.220.0.1 through 10.220.127.254

Table D-24 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-24 *Question 8: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	220	100	18
Mask	255	255	128	0
Subnet number	10	220	0	0
First address	10	220	0	1
Broadcast	10	220	127	255
Last Adress	10	220	127	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 128 = 128 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 128 that's closest to 100 but not bigger than 100. So, the third octet of the subnet number is 0.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 0 + 128 – 1 = 127.

This example tends to confuse people because a mask with 128 in it gives you subnet numbers that just do not seem to look right. Table D-25 gives you the answers for the first several subnets, just to make sure that you are clear about the subnets when using this mask with a Class A network.

Table D-25 *Question 8: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Zero Subnet	First Valid Subnet	Second Valid Subnet	Third Valid Subnet
Subnet	10.0.0.0	10.0.128.0	10.1.0.0	10.1.128.0
First address	10.0.0.1	10.0.128.1	10.1.0.1	10.1.128.1
Last address	10.0.127.254	10.0.255.254	10.1.127.254	10.1.255.254
Broadcast	10.0.127.255	10.0.255.255	10.1.127.255	10.1.255.255

Question 9: Answer

Table D-26 *Question 9: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.100.100	N/A
Mask	255.255.192.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	14	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 - 2 = 2$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{14} - 2 = 16,382$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-27. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-27 *Question 9: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.100.100	1010 1100 0001 1111 01**10 0100 0110 0100**
Mask	255.255.192.0	1111 1111 1111 1111 11**00 0000 0000 0000**
AND result (subnet number)	172.31.64.0	1010 1100 0001 1111 01**00 0000 0000 0000**
Change host to 1s (broadcast address)	172.31.127.255	1010 1100 0001 1111 01**11 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.64.1 through 172.31.127.254

Table D-28 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-28 *Question 9: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	100	100
Mask	255	255	192	0
Subnet number	172	31	64	0
First valid address	172	31	64	1
Broadcast	172	31	127	255
Last valid address	172	31	127	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 64 is the multiple of 64 that's closest to 100 but not bigger than 100. So, the third octet of the subnet number is 64.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 64 + 64 – 1 = 127.

Question 10: Answer

Table D-29 *Question 10: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.100.100	N/A
Mask	255.255.224.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	13	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 - 2 = 6$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{13} - 2 = 8190$	$2^{\text{number-of-host-bits}} - 2$

The binary calculations of the subnet number and broadcast address are in Table D-30. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-30 *Question 10: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.100.100	1010 1100 0001 1111 011**0 0100 0110 0100**
Mask	255.255.224.0	1111 1111 1111 1111 111**0 0000 0000 0000**
AND result (subnet number)	172.31.96.0	1010 1100 0001 1111 011**0 0000 0000 0000**
Change host to 1s (broadcast address)	172.31.127.255	1010 1100 0001 1111 011**1 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.96.1 through 172.31.127.254

Table D-31 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-31 *Question 10: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	100	100
Mask	255	255	224	0
Subnet number	172	31	96	0
First valid address	172	31	96	1
Broadcast	172	31	127	255
Last valid address	172	31	127	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 32 that's closest to 100 but not bigger than 100. So, the third octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 32 – 1 = 127.

Question 11: Answer

Table D-32 *Question 11: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.200.10	N/A
Mask	255.255.240.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	12	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 - 2 = 14$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{12} - 2 = 4094$	$2^{\text{number-of-host-bits}} - 2$

Table D-33 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-33 *Question 11: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.200.10	1010 1100 0001 1111 1100 **1000 0000 1010**
Mask	255.255.240.0	1111 1111 1111 1111 1111 **0000 0000 0000**
AND result (subnet number)	172.31.192.0	1010 1100 0001 1111 1100 **0000 0000 0000**
Change host to 1s (broadcast address)	172.31.207.255	1010 1100 0001 1111 1100 **1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.192.1 through 172.31.207.254

Table D-34 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-34 *Question 13: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	200	10
Mask	255	255	240	0
Subnet number	172	31	192	0
First valid address	172	31	192	1
Broadcast	172	31	207	255
Last valid address	172	31	207	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 16 that's closest to 200 but not bigger than 200. So, the third octet of the subnet number is 192.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 192 + 16 – 1 = 207.

Question 12: Answer

Table D-35 *Question 12: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.200.10	N/A
Mask	255.255.248.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	11	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 - 2 = 30$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{11} - 2 = 2046$	$2^{\text{number-of-host-bits}} - 2$

Table D-36 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-36 *Question 12: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.200.10	1010 1100 0001 1111 1100 1**000 0000 1010**
Mask	255.255.248.0	1111 1111 1111 1111 1111 1**000 0000 0000**
AND result (subnet number)	172.31.200.0	1010 1100 0001 1111 1100 1**000 0000 0000**
Change host to 1s (broadcast address)	172.31.207.255	1010 1100 0001 1111 1100 1**111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.200.1 through 172.31.207.254

Table D-37 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields

the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-37 *Question 12: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	200	10
Mask	255	255	248	0
Subnet number	172	31	200	0
First valid address	172	31	200	1
Broadcast	172	31	207	255
Last valid address	172	31	207	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 200 is the multiple of 8 that's closest to 200 but not bigger than 200. So, the third octet of the subnet number is 200.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 200 + 8 – 1 = 207.

Question 13: Answer

Table D-38 *Question 13: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.50.50	N/A
Mask	255.255.252.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	10	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 - 2 = 62$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{10} - 2 = 1022$	$2^{\text{number-of-host-bits}} - 2$

Table D-39 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-39 *Question 13: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.50.50	1010 1100 0001 1111 0011 00**10 0011 0010**
Mask	255.255.252.0	1111 1111 1111 1111 1111 11**00 0000 0000**
AND result (subnet number)	172.31.48.0	1010 1100 0001 1111 0011 00**00 0000 0000**
Change host to 1s (broadcast address)	172.31.51.255	1010 1100 0001 1111 0011 00**11 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.48.1 through 172.31.51.254

Table D-40 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-40 *Question 13: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	50	50
Mask	255	255	252	0
Subnet number	172	31	48	0
First valid address	172	31	48	1
Broadcast	172	31	51	255
Last valid address	172	31	51	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case,

48 is the multiple of 4 that's closest to 50 but not bigger than 50. So, the third octet of the subnet number is 48.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 48 + 4 – 1 = 51.

Question 14: Answer

Table D-41 *Question 14: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.50.50	N/A
Mask	255.255.254.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	9	Always defined as number of binary 0s in mask
Number of subnet bits	7	32 – (network size + host size)
Number of subnets	$2^7 - 2 = 126$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^9 - 2 = 510$	$2^{\text{number-of-host-bits}} - 2$

Table D-42 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-42 *Question 14: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.50.50	1010 1100 0001 1111 0011 001**0 0011 0010**
Mask	255.255.254.0	1111 1111 1111 1111 1111 111**0 0000 0000**
AND result (subnet number)	172.31.50.0	1010 1100 0001 1111 0011 001**0 0000 0000**
Change host to 1s (broadcast address)	172.31.51.255	1010 1100 0001 1111 0011 001**1 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.50.1 through 172.31.51.254

Table D-43 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-43 *Question 14: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	50	50
Mask	255	255	254	0
Subnet number	172	31	50	0
First valid address	172	31	50	1
Broadcast	172	31	51	255
Last valid address	172	31	51	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 254 = 2 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 50 is the multiple of 2 that's closest to 50 but not bigger than 50. So, the third octet of the subnet number is 50.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 50 + 2 – 1 = 51.

Question 15: Answer

Table D-44 *Question 15: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.140.14	N/A
Mask	255.255.255.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	8	Always defined as number of binary 0s in mask
Number of subnet bits	8	32 – (network size + host size)
Number of subnets	$2^8 - 2 = 254$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^8 - 2 = 254$	$2^{\text{number-of-host-bits}} - 2$

Table D-45 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-45 *Question 15: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.140.14	1010 1100 0001 1111 1000 1100 **0000 1110**
Mask	255.255.255.0	1111 1111 1111 1111 1111 1111 **0000 0000**
AND result (subnet number)	172.31.140.0	1010 1100 0001 1111 1000 1100 **0000 0000**
Change host to 1s (broadcast address)	172.31.140.255	1010 1100 0001 1111 1000 1100 **1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.140.1 through 172.31.140.254

Table D-46 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4.

Table D-46 *Question 15: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	140	14
Mask	255	255	255	0
Subnet number	172	31	140	0
First valid address	172	31	140	1
Broadcast	172	31	140	255
Last valid address	172	31	140	254

This subnetting scheme uses an easy mask because all of the octets are a 0 or a 255. No math tricks are needed at all!

Question 16: Answer

Table D-47 *Question 16: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.31.140.14	N/A
Mask	255.255.255.128	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	7	Always defined as number of binary 0s in mask
Number of subnet bits	9	32 – (network size + host size)
Number of subnets	$2^9 - 2 = 510$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^7 - 2 = 126$	$2^{\text{number-of-host-bits}} - 2$

Table D-48 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-48 *Question 16: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.31.140.14	1010 1100 0001 1111 1000 1100 0**000 1110**
Mask	255.255.255.128	1111 1111 1111 1111 1111 1111 1**000 0000**
AND result (subnet number)	172.31.140.0	1010 1100 0001 1111 1000 1100 0**000 0000**
Change host to 1s (broadcast address)	172.31.140.127	1010 1100 0001 1111 1000 1100 0**111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.31.140.1 through 172.31.140.126

Table D-49 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-49 *Question 16: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	31	140	14
Mask	255	255	255	128
Subnet number	172	31	140	0
First valid address	172	31	140	1
Broadcast	172	31	140	127
Last valid address	172	31	140	126

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 128 = 128 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 128 that's closest to 14 but not bigger than 14. So, the fourth octet of the subnet number is 0.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 0 + 128 – 1 = 127.

Question 17: Answer

Table D-50 *Question 17: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.15.150	N/A
Mask	255.255.255.192	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	6	Always defined as number of binary 0s in mask
Number of subnet bits	2	32 – (network size + host size)
Number of subnets	$2^2 - 2 = 2$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^6 - 2 = 62$	$2^{\text{number-of-host-bits}} - 2$

Table D-51 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find

the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-51 *Question 17: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.150	1100 0000 1010 1000 0000 1111 10**01 0110**
Mask	255.255.255.192	1111 1111 1111 1111 1111 1111 11**00 0000**
AND result (subnet number)	192.168.15.128	1100 0000 1010 1000 0000 1111 10**00 0000**
Change host to 1s (broadcast address)	192.168.15.191	1100 0000 1010 1000 0000 1111 10**11 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.129 through 192.168.15.190

Table D-52 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-52 *Question 17: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	15	150
Mask	255	255	255	192
Subnet number	192	168	15	128
First valid address	192	168	15	129
Broadcast	192	168	15	191
Last valid address	192	168	15	190

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 64 that's closest to 150 but not bigger than 150. So, the fourth octet of the subnet number is 128.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 128 + 64 – 1 = 191.

Question 18: Answer

Table D-53 *Question 18: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.15.150	N/A
Mask	255.255.255.224	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	5	Always defined as number of binary 0s in mask
Number of subnet bits	3	32 – (network size + host size)
Number of subnets	$2^3 - 2 = 6$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^5 - 2 = 30$	$2^{\text{number-of-host-bits}} - 2$

Table D-54 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-54 *Question 18: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.150	1100 0000 1010 1000 0000 1111 100**1 0110**
Mask	255.255.255.224	1111 1111 1111 1111 1111 1111 111**0 0000**
AND result (subnet number)	192.168.15.128	1100 0000 1010 1000 0000 1111 100**0 0000**
Change host to 1s (broadcast address)	192.168.15.159	1100 0000 1010 1000 0000 1111 100**1 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.129 through 192.168.15.158

Table D-55 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-55 *Question 18: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	15	150
Mask	255	255	255	224
Subnet number	192	168	15	128
First valid address	192	168	15	129
Broadcast	192	168	15	159
Last valid address	192	168	15	158

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 224 = 32 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 128 is the multiple of 32 that's closest to 150 but not bigger than 150. So, the fourth octet of the subnet number is 128.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 128 + 32 – 1 = 159.

Question 19: Answer

Table D-56 *Question 19: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.100.100	N/A
Mask	255.255.255.240	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	4	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 - 2 = 14$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^4 - 2 = 14$	$2^{\text{number-of-host-bits}} - 2$

Table D-57 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-57 *Question 19: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.100.100	1100 0000 1010 1000 0110 0100 0110 **0100**
Mask	255.255.255.240	1111 1111 1111 1111 1111 1111 1111 **0000**
AND result (subnet number)	192.168.100.96	1100 0000 1010 1000 0110 0100 0110 **0000**
Change host to 1s (broadcast address)	192.168.100.111	1100 0000 1010 1000 0110 0100 0110 **1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.100.97 through 192.168.100.110

Table D-58 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-58 *Question 19: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	**Octet 2**	**Octet 3**	**Octet 4**
Address	192	168	100	100
Mask	255	255	255	240
Subnet number	192	168	100	96
First valid address	192	168	100	97
Broadcast	192	168	100	111
Last valid address	192	168	100	110

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 16 that's closest to 100 but not bigger than 100. So, the fourth octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 16 – 1 = 111.

Question 20: Answer

Table D-59 *Question 20: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.100.100	N/A
Mask	255.255.255.248	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	3	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 - 2 = 30$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^3 - 2 = 6$	$2^{\text{number-of-host-bits}} - 2$

Table D-60 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-60 *Question 20: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.100.100	1100 0000 1010 1000 0110 0100 0110 0**100**
Mask	255.255.255.248	1111 1111 1111 1111 1111 1111 1111 1**000**
AND result (subnet number)	192.168.100.96	1100 0000 1010 1000 0110 0100 0110 0**000**
Change host to 1s (broadcast address)	192.168.100.103	1100 0000 1010 1000 0110 0100 0110 0**111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.100.97 through 192.168.100.102

Table D-61 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-61 *Question 20: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	100	100
Mask	255	255	255	248
Subnet number	192	168	100	96
First valid address	192	168	100	97
Broadcast	192	168	100	103
Last valid address	192	168	100	102

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 96 is the multiple of 8 that's closest to 100 but not bigger than 100. So, the fourth octet of the subnet number is 96.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 96 + 8 – 1 = 103.

Question 21: Answer

Table D-62 *Question 21: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	192.168.15.230	N/A
Mask	255.255.255.252	N/A
Number of network bits	24	Always defined by Class A, B, C
Number of host bits	2	Always defined as number of binary 0s in mask
Number of subnet bits	6	32 – (network size + host size)
Number of subnets	$2^6 - 2 = 62$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^2 - 2 = 2$	$2^{\text{number-of-host-bits}} - 2$

Table D-63 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-63 *Question 21: Binary Calculation of Subnet and Broadcast Addresses*

Address	192.168.15.230	1100 0000 1010 1000 0000 1111 1110 01**10**
Mask	255.255.255.252	1111 1111 1111 1111 1111 1111 1111 11**00**
AND result (subnet number)	192.168.15.228	1100 0000 1010 1000 0000 1111 1110 01**00**
Change host to 1s (broadcast address)	192.168.15.231	1100 0000 1010 1000 0000 1111 1110 01**11**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

192.168.15.229 through 192.168.15.230

Table D-64 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-64 *Question 21: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	192	168	15	230
Mask	255	255	255	252
Subnet number	192	168	15	228
First valid address	192	168	15	229
Broadcast	192	168	15	231
Last valid address	192	168	15	230

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 252 = 4 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 228 is the multiple of 4 that's closest to 230 but not bigger than 230. So, the fourth octet of the subnet number is 228.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet

number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 228 + 4 – 1 = 231.

Question 22: Answer

Table D-65 *Question 22: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.1.1.1	N/A
Mask	255.248.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	19	Always defined as number of binary 0s in mask
Number of subnet bits	5	32 – (network size + host size)
Number of subnets	$2^5 - 2 = 30$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{19} - 2 = 524{,}286$	$2^{\text{number-of-host-bits}} - 2$

Table D-66 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-66 *Question 22: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.1.1.1	0000 1010 0000 0**001 0000 0001 0000 0001**
Mask	255.248.0.0	1111 1111 1111 1**000 0000 0000 0000 0000**
AND result (subnet number)	10.0.0.0	0000 1010 0000 0**000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.7.255.255	0000 1010 0000 0**111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.0.0.1 through 10.7.255.254

Take a closer look at the subnet part of the subnet address, as is shown in bold here: 0000 1010 **0000 0**000 0000 0000 0000 0000. The subnet part of the address is all binary 0s, making this subnet a zero subnet.

Table D-67 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-67 *Question 22: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	1	1	1
Mask	255	248	0	0
Subnet number	10	0	0	0
First valid address	10	0	0	1
Broadcast	10	7	255	255
Last valid address	10	7	255	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The second octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 248 = 8 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 8 that's closest to 1 but not bigger than 1. So, the second octet of the subnet number is 0.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 0 + 8 – 1 = 7.

Question 23: Answer

Table D-68 *Question 23: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.16.1.200	N/A
Mask	255.255.240.0	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	12	Always defined as number of binary 0s in mask
Number of subnet bits	4	32 – (network size + host size)
Number of subnets	$2^4 - 2 = 14$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{12} - 2 = 4094$	$2^{\text{number-of-host-bits}} - 2$

Table D-69 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-69 *Question 23: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.16.1.200	1010 1100 0001 0000 0000 **0001 1100 1000**
Mask	255.255.240.0	1111 1111 1111 1111 1111 **0000 0000 0000**
AND result (subnet number)	172.16.0.0	1010 1100 0001 0000 0000 **0000 0000 0000**
Change host to 1s (broadcast address)	172.16.15.255	1010 1100 0001 0000 0000 **1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.16.0.1 through 172.16.15.254

Take a closer look at the subnet part of the subnet address, as shown in bold here: 1010 1100 0001 0000 **0000** 0000 0000 0000. The subnet part of the address is all binary 0s, making this subnet a zero subnet.

Table D-70 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-70 *Question 23: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	16	1	200
Mask	255	255	240	0
Subnet number	172	16	0	0
First valid address	172	16	0	1
Broadcast	172	16	15	255
Last valid address	172	16	15	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The third octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 240 = 16 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 0 is the multiple of 16 that's closest to 1 but not bigger than 1. So, the third octet of the subnet number is 0.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 0 + 16 – 1 = 15.

Question 24: Answer

Table D-71 *Question 24: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	172.16.0.200	N/A
Mask	255.255.255.192	N/A
Number of network bits	16	Always defined by Class A, B, C
Number of host bits	6	Always defined as number of binary 0s in mask
Number of subnet bits	10	32 – (network size + host size)
Number of subnets	$2^{10} - 2 = 1022$	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^6 - 2 = 62$	$2^{\text{number-of-host-bits}} - 2$

Table D-72 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-72 *Question 24: Binary Calculation of Subnet and Broadcast Addresses*

Address	172.16.0.200	1010 1100 0001 0000 0000 0000 11**00 1000**
Mask	255.255.255.192	1111 1111 1111 1111 1111 1111 11**00 0000**
AND result (subnet number)	172.16.0.192	1010 1100 0001 0000 0000 0000 11**00 0000**
Change host to 1s (broadcast address)	172.16.0.255	1010 1100 0001 0000 0000 0000 11**11 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

172.16.0.193 through 172.16.0.254

Table D-73 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4. Remember, subtracting the interesting (non-0 or 255) mask value from 256 yields the magic number. The magic number multiple that's closest to but not larger than the IP address's interesting octet value is the subnet value in that octet.

Table D-73 *Question 24: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	172	16	0	200
Mask	255	255	255	192
Subnet number	172	16	0	192
First valid address	172	16	0	193
Broadcast	172	16	0	255
Last valid address	172	16	0	254

This subnetting scheme uses a hard mask because one of the octets is not a 0 or a 255. The fourth octet is "interesting" in this case. The key part of the trick to get the right answers is to calculate the magic number, which is 256 – 192 = 64 in this case (256 – mask's value in the interesting octet). The subnet number's value in the interesting octet (inside the box) is the multiple of the magic number that's not bigger than the original IP address's value in the interesting octet. In this case, 192 is the multiple of 64 that's closest to 200 but not bigger than 200. So, the fourth octet of the subnet number is 192.

The second tricky part of this process calculates the subnet broadcast address. The full process is described in Chapter 4, but the tricky part is, as usual, in the "interesting" octet. Take the subnet number's value in the interesting octet, add the magic number, and subtract 1. That's the broadcast address's value in the interesting octet. In this case, 192 + 64 – 1 = 255.

You can easily forget that the subnet part of this address, when using this mask, actually covers all of the third octet as well as 2 bits of the fourth octet. For instance, the valid subnet numbers in order are listed here, starting with the first valid subnet by avoiding subnet 172.16.0.0–the zero subnet in this case:

172.16.0.64
172.16.0.128
172.16.0.192
172.16.1.0

172.16.1.64
172.16.1.128
172.16.1.192
172.16.2.0
172.16.2.64
172.16.2.128
172.16.2.192
172.16.3.0
172.16.3.64
172.16.3.128
172.16.3.192

And so on.

Question 25: Answer

Congratulations, you made it through all the extra subnetting practice! Here's an easy one to complete your review—one with no subnetting at all!

Table D-74 *Question 25: Size of Network, Subnet, Host, Number of Subnets, Number of Hosts*

Step	Example	Rules to Remember
Address	10.1.1.1	N/A
Mask	255.0.0.0	N/A
Number of network bits	8	Always defined by Class A, B, C
Number of host bits	24	Always defined as number of binary 0s in mask
Number of subnet bits	0	32 – (network size + host size)
Number of subnets	0	$2^{\text{number-of-subnet-bits}} - 2$
Number of hosts	$2^{24} - 2 =$ 16,777,214	$2^{\text{number-of-host-bits}} - 2$

Table D-75 shows the binary calculations of the subnet number and broadcast address. To calculate the subnet number, perform a Boolean AND of the address with the subnet mask. To find the broadcast address for this subnet, change all the host bits to binary 1s in the subnet number. The host bits are in **bold** print in the table.

Table D-75 *Question 25: Binary Calculation of Subnet and Broadcast Addresses*

Address	10.1.1.1	0000 1010 **0000 0001 0000 0001 0000 0001**
Mask	255.0.0.0	1111 1111 **0000 0000 0000 0000 0000 0000**
AND result (subnet number)	10.0.0.0	0000 1010 **0000 0000 0000 0000 0000 0000**
Change host to 1s (broadcast address)	10.255.255.255	0000 1010 **1111 1111 1111 1111 1111 1111**

Just add 1 to the subnet number to get the first valid IP address; just subtract 1 from the broadcast address to get the last valid IP address. In this case:

10.0.0.1 through 10.255.255.254

Table D-76 lists the way to get the same answers using the subnet chart and magic math described in Chapter 4.

Table D-76 *Question 25: Subnet, Broadcast, First and Last Addresses Calculated Using Subnet Chart*

	Octet 1	Octet 2	Octet 3	Octet 4
Address	10	1	1	1
Mask	255	0	0	0
Network number	10	0	0	0
First valid address	10	0	0	1
Broadcast	10	255	255	255
Last valid address	10	255	255	254

Discovering All Subnets When Using SLSM: 13 Questions

This section covers the second class of IP addressing problems mentioned in the introduction to this appendix. The question is as follows:

Assuming SLSM, what are the subnets of this network?

For practice, answer that question for the following networks and masks:

1. 10.0.0.0, mask 255.192.0.0
2. 10.0.0.0, mask 255.224.0.0
3. 10.0.0.0, mask 255.248.0.0
4. 10.0.0.0, mask 255.252.0.0
5. 10.0.0.0, mask 255.255.128.0
6. 10.0.0.0, mask 255.255.192.0
7. 172.31.0.0, mask 255.255.224.0
8. 172.31.0.0, mask 255.255.240.0
9. 172.31.0.0, mask 255.255.252.0
10. 172.31.0.0, mask 255.255.255.224
11. 192.168.15.0, mask 255.255.255.192

12. 192.168.15.0, mask 255.255.255.224

13. 192.168.15.0, mask 255.255.255.240

These questions are mostly a subset of the same 25 subnetting questions covered in the first section of this appendix. The explanations of the answers will be based on the seven-step algorithm from Chapter 4, repeated here for convenience. Also, keep in mind that this formal algorithm assumes that the subnet field is 8 bits in length or less. However, some problems in this appendix have a longer subnet field. For those problems, the answer explains how to expand the logic in this baseline algorithm.

Step 1 Write down the classful network number.

Step 2 For the first (lowest numeric) subnet number, copy the entire network number. That is the first subnet number, and is also the zero subnet.

Step 3 Decide which octet contains the entire subnet field; call this octet the interesting octet. (Remember, this algorithm assumes 8 subnet bits or less.)

Step 4 Calculate the magic number by subtracting the mask's interesting octet value from 256.

Step 5 Copy down the previous subnet number's noninteresting octets onto the next line as the next subnet number; only one octet is missing at this point.

Step 6 Add the magic number to the previous subnet's interesting octet, and write that down as the next subnet number's interesting octet, completing the next subnet number.

Step 7 Repeat Steps 5 and 6 until the new interesting octet is 256. That subnet is not valid. The previously calculated subnet is the last valid subnet, and also the broadcast subnet.

Question 1: Answer

This question begins with the following basic facts:

Network 10.0.0.0
Mask 255.192.0.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 2nd
Magic number: 256 – 192 = 64

From there, Table D-77 shows the rest of the steps for the process.

Table D-77 *Question 1 Answer: Network 10.0.0.0, Mask 255.192.0.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	10	0	0	0
2) Zero subnet	10	0	0	0
5) Next subnet	10	64	0	0
6) Next subnet	10	128	0	0
6) Broadcast subnet	10	192	0	0
7) Invalid subnet*	10	256	0	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the broadcast subnet number may not have been obvious until attempting to write the final (invalid) next subnet number, as seen in the last row of the table. You can follow the steps shown in the table, knowing that when the interesting octet's value is 256, you have gone too far. The broadcast subnet is the subnet that was found one step prior.

Alternately, you can find the broadcast subnet based on the following fact: the broadcast subnet's interesting octet is equal to the subnet mask value in that same octet.

Question 2: Answer

This question begins with the following basic facts:

Network 10.0.0.0
Mask 255.224.0.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 2nd
Magic number: 256 – 224 = 32

From there, Table D-78 shows the rest of the steps for the process.

Table D-78 *Question 2 Answer: Network 10.0.0.0, Mask 255.224.0.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	10	0	0	0
2) Zero subnet	10	0	0	0
5) Next subnet	10	32	0	0
6) Next subnet	10	64	0	0
6) Next subnet	10	96	0	0
6) Generic representation of next subnet	10	X	0	0
6) Broadcast subnet	10	224	0	0
7) Invalid subnet*	10	256	0	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (second octet in this case) simply increments by the magic number. To reduce the space required by the table, after the pattern is obvious, the table represents the remaining subnet numbers before the broadcast subnet as a generic value, 10.X.0.0. The subnets not specifically listed are 10.128.0.0, 10.160.0.0, and 10.192.0.0.

Question 3: Answer

This question begins with the following basic facts:

Network 10.0.0.0
Mask 255.248.0.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 2nd
Magic number: 256 – 248 = 8

From there, Table D-79 shows the rest of the steps for the process.

Table D-79 *Question 3 Answer: Network 10.0.0.0, Mask 255.248.0.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	10	0	0	0
2) Zero subnet	10	0	0	0
5) Next subnet	10	8	0	0

continues

Table D-79 *Question 3 Answer: Network 10.0.0.0, Mask 255.248.0.0 (Continued)*

Step	Octet #1	Octet #2	Octet #3	Octet #4
6) Next subnet	10	16	0	0
6) Next subnet	10	24	0	0
6) Generic representation of next subnet	10	X	0	0
6) Broadcast subnet	10	248	0	0
7) Invalid subnet*	10	256	0	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (second octet in this case) simply increments by the magic number. To reduce the space required by the table, after the pattern is obvious, the table represents the remaining subnet numbers before the broadcast subnet as a generic value, 10.x.0.0. The subnets not specifically listed simply have a multiple of 8 in the second octet.

Question 4: Answer

This question begins with the following basic facts:

Network 10.0.0.0
Mask 255.252.0.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 2^{nd}
Magic number: 256 – 252 = 4

From there, Table D-80 shows the rest of the steps for the process.

Table D-80 *Question 4 Answer: Network 10.0.0.0, Mask 255.252.0.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	10	0	0	0
2) Zero subnet	10	0	0	0
5) Next subnet	10	4	0	0
6) Next subnet	10	8	0	0
6) Next subnet	10	12	0	0

Table D-80 *Question 4 Answer: Network 10.0.0.0, Mask 255.252.0.0 (Continued)*

Step	Octet #1	Octet #2	Octet #3	Octet #4
6) Generic representation of next subnet	10	X	0	0
6) Broadcast subnet	10	252	0	0
7) Invalid subnet*	10	256	0	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (second octet in this case) simply increments by the magic number. To reduce the space required by the table, after the pattern is obvious, the table represents the remaining subnet numbers before the broadcast subnet as a generic value, 10.x.0.0. The subnets not specifically listed simply have a multiple of 4 in the second octet.

Question 5: Answer

This question begins with the following basic facts:

Network 10.0.0.0
Mask 255.255.128.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 3rd
Magic number: 256 – 128 = 128

This question actually uses a subnet field that spans all of the second octet, and a single bit in the third octet. As a result, the original seven-step process, which assumes a 1-octet-or-less subnet field, cannot be used. However, an expanded process is described along with the answer to this question.

> **NOTE** Many of you may intuitively see the way to find the complete answer to this question, long before you finish reading the revised process listed here. If you think you are getting the idea, you probably are, so do not let the details in the text get in the way.

First, Table D-81 shows the beginning of the process, which occurs just like the earlier examples, except that the interesting octet is now the third octet.

Table D-81 *Question 5 Answer, Part 1: Network 10.0.0.0, Mask 255.255.128.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	10	0	0	0
2) Zero subnet	10	0	0	0
5) Next subnet	10	0	128	0
7) Invalid subnet*	10	0	256	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

At this point, the last number is obviously an invalid subnet number due to the 256 in the third octet. Instead of that fact signifying the end of the process, it means you should do the following:

> Record the next subnet, based on the following changes to the previous valid subnet number: add 1 to the octet to the left of the interesting octet, and set the interesting octet to 0.

In this case, this new step runs as follows:

- The previous valid subnet is 10.0.128.0.
- Add 1 to the octet to the left of the interesting octet (value 0); the next subnet number's second octet will then be 1.
- The next subnet number's interesting octet will be 0.

Each time the next subnet number would have had a 256 in the interesting octet, you instead follow this new step. It is a little like normal decimal addition. For example, when you add 319 and 1, you add 1 and 9, write down a 0, and carry the 1 to the next digit to the left. It is much more obvious through examples, though. So, to complete the logic, Table D-82 shows the example, with this new logic implemented. (Note that the new step has been labeled as Step 8.)

Table D-82 *Question 5 Answer, Part 2: Network 10.0.0.0, Mask 255.255.128.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	10	0	0	0
2) Zero subnet	10	0	0	0
5) Next subnet	10	0	128	0
8) Increment in the octet to the left, and use 0 in the interesting octet	10	1	0	0

Table D-82 *Question 5 Answer, Part 2: Network 10.0.0.0, Mask 255.255.128.0 (Continued)*

Step	Octet #1	Octet #2	Octet #3	Octet #4
5) Next subnet	10	1	128	0
8) Increment in the octet to the left, and use 0 in the interesting octet	10	2	0	0
5) Next subnet	10	2	128	0
8) Increment in the octet to the left, and use 0 in the interesting octet	10	3	0	0
5) Next subnet	10	3	128	0
8) Increment in the octet to the left, and use 0 in the interesting octet	10	4	0	0
5) Next subnet	10	4	128	0
5) Generic view	10	X	0/128	0
6) Broadcast subnet	10	255	128	0
7) Invalid subnet*	10	256	0	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

The end of the table is found in this example when the octet to the left of the interesting octet reaches 256. The previously listed subnet is the broadcast subnet.

Question 6: Answer

This question begins with the following basic facts:

Network 10.0.0.0
Mask 255.255.192.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 3^{rd}
Magic number: 256 – 192 = 64

Like the previous question, this question actually uses a subnet field larger than 1 octet. As a result, the expanded version of the seven-step process is used. First, Table D-83 shows the beginning of the process, which occurs just like the standard seven-step process.

Table D-83 *Question 6 Answer, Part 1: Network 10.0.0.0, Mask 255.255.192.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	10	0	0	0
2) Zero subnet	10	0	0	0
5) Next subnet	10	0	64	0
5) Next subnet	10	0	128	0
5) Next subnet	10	0	192	0
7) Invalid subnet*	10	0	256	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

After finding a 256 in the interesting octet, the extra bit of logic is applied, as follows:

> Record the next subnet, based on the following changes to the previous valid subnet number: add 1 to the octet to the left of the interesting octet, and set the interesting octet to 0.

Table D-84 shows the actual values.

Table D-84 *Question 6 Answer, Part 2: Network 10.0.0.0, Mask 255.255.192.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	10	0	0	0
2) Zero subnet	10	0	0	0
5) Next subnet	10	0	64	0
5) Next subnet	10	0	128	0
5) Next subnet	10	0	192	0
8) Increment in the octet to the left, and use 0 in the interesting octet	10	1	0	0
5) Next subnet	10	1	64	0
5) Next subnet	10	1	128	0
5) Next subnet	10	1	192	0
8) Increment in the octet to the left, and use 0 in the interesting octet	10	2	0	0
5) Next subnet	10	2	64	0

Table D-84 *Question 6 Answer, Part 2: Network 10.0.0.0, Mask 255.255.192.0 (Continued)*

Step	Octet #1	Octet #2	Octet #3	Octet #4
5) Generic view	10	X	0/64/128/ 192	0
6) Broadcast subnet	10	255	192	0
7) Invalid subnet*	10	256	0	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

The end of the table is found in this example when the octet to the left of the interesting octet reaches 256. The previously listed subnet is the broadcast subnet.

Question 7: Answer

This question begins with the following basic facts:

Network 172.31.0.0
Mask 255.255.224.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 3^{rd}
Magic number: 256 – 224 = 32

From there, Table D-85 shows the rest of the steps for the process.

Table D-85 *Question 7 Answer: Network 172.31.0.0, Mask 255.255.224.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	172	31	0	0
2) Zero subnet	172	31	0	0
5) Next subnet	172	31	32	0
5) Next subnet	172	31	64	0
5) Next subnet	172	31	96	0
5) Next subnet	172	31	128	0

continues

Table D-85 *Question 7 Answer: Network 172.31.0.0, Mask 255.255.224.0 (Continued)*

Step	Octet #1	Octet #2	Octet #3	Octet #4
5) Next subnet	172	31	160	0
5) Next subnet	172	31	192	0
6) Broadcast subnet	172	31	224	0
7) Invalid subnet*	172	31	256	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (third octet in this case) simply increments by the magic number.

Question 8: Answer

This question begins with the following basic facts:

Network 172.31.0.0
Mask 255.255.240.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 3rd
Magic number: 256 – 240 = 16

From there, Table D-86 shows the rest of the steps for the process.

Table D-86 *Question 8 Answer: Network 172.31.0.0, Mask 255.255.240.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	172	31	0	0
2) Zero subnet	172	31	0	0
5) Next subnet	172	31	16	0
5) Next subnet	172	31	32	0
5) Next subnet	172	31	48	0
5) Next subnet	172	31	64	0

Table D-86 *Question 8 Answer: Network 172.31.0.0, Mask 255.255.240.0 (Continued)*

Step	Octet #1	Octet #2	Octet #3	Octet #4
5) Next subnet	172	31	x	0
6) Broadcast subnet	172	31	240	0
7) Invalid subnet*	172	31	256	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (third octet in this case) simply increments by the magic number. To reduce the space required by the table, the table represents the remaining subnet numbers before the broadcast subnet as a generic value, 172.31.x.0. The subnets not specifically listed simply have a multiple of 16 in the third octet.

Question 9: Answer

This question begins with the following basic facts:

Network 172.31.0.0
Mask 255.255.252.0

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 3rd
Magic number: 256 – 252 = 4

From there, Table D-87 shows the rest of the steps for the process.

Table D-87 *Question 9 Answer: Network 172.31.0.0, Mask 255.255.252.0*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	172	31	0	0
2) Zero subnet	172	31	0	0
5) Next subnet	172	31	4	0
5) Next subnet	172	31	8	0
5) Next subnet	172	31	12	0
5) Next subnet	172	31	16	0
5) Next subnet	172	31	x	0
6) Broadcast subnet	172	31	252	0
7) Invalid subnet*	172	31	256	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (third octet in this case) simply increments by the magic number. To reduce the space required by the table, the table represents the remaining subnet numbers before the broadcast subnet as a generic value, 172.31.x.0. The subnets not specifically listed simply have a multiple of 4 in the third octet.

Question 10: Answer

This question begins with the following basic facts:

Network 172.31.0.0
Mask 255.255.255.224

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 4th
Magic number: 256 – 224 = 32

This question uses a subnet field larger than 1 octet, requiring the expanded version of the process as seen in Questions 5 and 6. Table D-88 shows the beginning of the process.

Table D-88 *Question 10 Answer, Part 1: Network 172.31.0.0, Mask 255.255.255.224*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	172	31	0	0
2) Zero subnet	172	31	0	0
5) Next subnet	172	31	0	32
5) Next subnet	172	31	0	64
5) Next subnet	172	31	0	96
5) Next subnet	172	31	0	128
5) Next subnet	172	31	0	160
5) Next subnet	172	31	0	192
5) Next subnet	172	31	0	224
7) Invalid subnet*	172	31	0	256

*The invalid subnet row is just a reminder used by this process as to when to stop.

After finding a 256 in the interesting octet, the extra bit of logic is applied, as follows:

> Record the next subnet, based on the following changes to the previous valid subnet number: add 1 to the octet to the left of the interesting octet, and set the interesting octet to 0.

Table D-89 shows the actual values.

Table D-89 *Question 10 Answer, Part 2: Network 172.31.0.0, Mask 255.255.255.224*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	172	31	0	0
2) Zero subnet	172	31	0	0
5) Next subnet	172	31	0	32
5) Next subnet	172	31	0	64
5) Next subnet	172	31	0	128
5) Next subnet	172	31	0	192
5) Next subnet	172	31	0	224
8) Increment in the octet to the left, and use 0 in the interesting octet	172	31	1	0
5) Next subnet	172	31	1	32
5) Next subnet	172	31	1	64
5) Next subnet	172	31	1	128
5) Next subnet	172	31	1	160
5) Next subnet	172	31	1	192
5) Next subnet	172	31	1	224
8) Increment in the octet to the left, and use 0 in the interesting octet	172	31	2	0
5) Generic view	172	31	X	Y
6) Broadcast subnet	172	31	255	224
7) Invalid subnet*	172	31	256	0

*The invalid subnet row is just a reminder used by this process as to when to stop.

The end of the table is found in this example when the octet to the left of the interesting octet reaches 256. The previously listed subnet is the broadcast subnet.

Question 11: Answer

This question begins with the following basic facts:

Network 192.168.15.0
Mask 255.255.255.192

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 4th
Magic number: 256 – 192 = 64

From there, Table D-90 shows the rest of the steps for the process.

Table D-90 *Question 11 Answer: Network 192.168.15.0, Mask 255.255.255.192*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	192	168	15	0
2) Zero subnet	192	168	15	0
5) Next subnet	192	168	15	64
5) Next subnet	192	168	15	128
6) Broadcast subnet	192	168	15	192
7) Invalid subnet*	192	168	15	256

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (fourth octet in this case) simply increments by the magic number.

Question 12: Answer

This question begins with the following basic facts:

Network 192.168.15.0
Mask 255.255.255.224

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 4th
Magic number: 256 – 224 = 32

From there, Table D-91 shows the rest of the steps for the process.

Table D-91 *Question 11 Answer: Network 192.168.15.0, Mask 255.255.255.224*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	192	168	15	0
2) Zero subnet	192	168	15	0
5) Next subnet	192	168	15	32
5) Next subnet	192	168	15	64
5) Next subnet	192	168	15	96
5) Generic view	192	168	15	X
6) Broadcast subnet	192	168	15	224
7) Invalid subnet*	192	168	15	256

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (fourth octet in this case) simply increments by the magic number. To reduce the space required by the table, the table represents the remaining subnet numbers before the broadcast subnet as a generic value, 192.168.15.x. The subnets not specifically listed simply have a multiple of 32 in the fourth octet.

Question 13: Answer

This question begins with the following basic facts:

Network 192.168.15.0
Mask 255.255.255.240

From there, Steps 3 and 4 ask for the following pieces of information:

Interesting octet: 4^{th}
Magic number: 256 – 240 = 16

From there, Table D-92 shows the rest of the steps for the process.

Table D-92 *Question 13 Answer: Network 192.168.15.0, Mask 255.255.255.240*

Step	Octet #1	Octet #2	Octet #3	Octet #4
1) Network number	192	168	15	0
2) Zero subnet	192	168	15	0
5) Next subnet	192	168	15	16
5) Next subnet	192	168	15	32
5) Next subnet	192	168	15	48
5) Generic view	192	168	15	X
6) Broadcast subnet	192	168	15	240
7) Invalid subnet*	192	168	15	256

*The invalid subnet row is just a reminder used by this process as to when to stop.

Note that the subnet numbers' interesting octet (fourth octet in this case) simply increments by the magic number. To reduce the space required by the table, the table represents the remaining subnet numbers before the broadcast subnet as a generic value, 192.168.15.x. The subnets not specifically listed simply have a multiple of 16 in the fourth octet.

Discovering the Smallest Inclusive Summary Route: 10 Questions

The last two major sections of this appendix provide practice questions to find the best inclusive and exclusive summary routes, respectively. For the following ten lists of subnets, discover the subnet/mask or prefix/length for the smallest possible inclusive summary route:

1. 10.20.30.0/24, 10.20.40.0/24, 10.20.35.0/24, 10.20.45.0/24
2. 10.20.7.0/24, 10.20.4.0/24, 10.20.5.0/24, 10.20.6.0/24
3. 10.20.3.0/24, 10.20.4.0/24, 10.20.5.0/24, 10.20.6.0/24, 10.20.7.0/24, 10.20.8.0/24
4. 172.16.200.0/23, 172.16.204.0/23, 172.16.208.0/23
5. 172.16.200.0/23, 172.16.204.0/23, 172.16.208.0/23, 172.16.202.0/23, 172.16.206.0/23
6. 172.16.120.0/22, 172.16.112.0/22, 172.16.124.0/22, 172.16.116.0/22
7. 192.168.1.16/29, 192.168.1.32/29, 192.168.1.24/29

8. 192.168.1.16/29, 192.168.1.32/29
9. 10.1.80.0/25, 10.1.81.0/25, 10.1.81.128/25
10. 10.1.80.0/26, 10.1.81.0/26, 10.1.81.128/26

The following steps are a repeat of the algorithm found in Chapter 4. Chapter 4 only explained details assuming consecutive subnets and SLSM, but the algorithm works fine with SLSM or VLSM, and with nonconsecutive subnets. However, nonconsecutive subnets typically require more passes through the algorithm logic. If VLSM is used, at Step 2, you subtract *y* from the longest prefix length to start the process, again requiring many more steps through the process.

Step 1 Count the number of subnets; then, find the smallest value of *y*, such that 2^y => that number of subnets.

Step 2 For the next step, use a prefix length of the prefix length for each of the component subnets, minus *y*.

Step 3 Pretend that the lowest subnet number in the list of component subnets is an IP address. Using the new, smaller prefix from Step 2, calculate the subnet number in which this pretend address resides.

Step 4 Repeat Step 3 for the largest numeric component subnet number and the same prefix. If it is the same subnet derived as in Step 3, the resulting subnet is the best summarized route, using the new prefix.

Step 5 If Steps 3 and 4 do not yield the same resulting subnet, repeat Steps 3 and 4, with another new prefix length of 1 less than the last prefix length.

Question 1: Answer

This question begins with the following routes that need to be summarized:

10.20.30.0/24
10.20.35.0/24
10.20.40.0/24
10.20.45.0/24

The first two steps are as follows:

1) Y = 2, because there are 4 component routes, and 2^2 => 4
2) Start with a prefix length of 24 – 2 = 22

From there, Table D-93 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the two steps match.

Table D-93 *Question 1 Answer: Inclusive Summary of 4 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
22	10.20.30.0/22 yields a subnet of 10.20.28.0/22	10.20.45.0/22 yields a subnet of 10.20.44.0/22
21	10.20.30.0/21 yields a subnet of 10.20.24.0/21	10.20.45.0/21 yields a subnet of 10.20.40.0/21
20	10.20.30.0/20 yields a subnet of 10.20.16.0/20	10.20.45.0/20 yields a subnet of 10.20.32.0/20
19	10.20.30.0/19 yields a subnet of 10.20.0.0/19	10.20.45.0/19 yields a subnet of 10.20.32.0/19
18	10.20.30.0/18 yields a subnet of 10.20.0.0/18	10.20.45.0/18 yields a subnet of 10.20.0.0/18

This question requires that you iterate through several progressively shorter prefix lengths until you find the correct answer. Finally, the process shows that 10.20.0.0/18 would be the smallest inclusive summary. For questions in which the component subnets are not consecutive, as was the case in this question, you might try to guess a better starting point for the prefix length (a few bits shorter) rather than starting with Steps 1 and 2 of the stated process. Regardless, the process will give you the right answer.

Question 2: Answer

This question begins with the following routes that need to be summarized:

10.20.4.0/24
10.20.5.0/24
10.20.6.0/24
10.20.7.0/24

The first two steps are as follows:

1) Y = 2, because there are 4 component routes, and 2^2 => 4
2) Start with a prefix length of 24 – 2 = 22

From there, Table D-94 shows the iterations through Steps 3 and 4. Remember, you do the math using the original smallest and largest component subnets as if they were IP addresses, using progressively shorter prefix lengths, until the results are the same. If the results are the same, then you have found the smallest inclusive summary.

Table D-94 *Question 2 Answer: Inclusive Summary of 4 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
22	10.20.4.0/22 yields a subnet of 10.20.4.0/22	10.20.7.0/22 yields a subnet of 10.20.4.0/22

Question 3: Answer

This question begins with the following routes that need to be summarized:

10.20.3.0/24
10.20.4.0/24
10.20.5.0/24
10.20.6.0/24
10.20.7.0/24
10.20.8.0/24

The first two steps are as follows:

1) Y = 3, because there are 6 component routes, and 2^3 => 6
2) Start with a prefix length of 24 – 3 = 21

From there, Table D-95 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the right answer is found.

Table D-95 *Question 3 Answer: Inclusive Summary of 6 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
21	10.20.3.0/21 yields a subnet of 10.20.0.0/21	10.20.8.0/21 yields a subnet of 10.20.8.0/21
20	10.20.3.0/20 yields a subnet of 10.20.0.0/20	10.20.8.0/20 yields a subnet of 10.20.0.0/20

After two passes through Steps 3 and 4, the results are equal, implying that 10.20.0.0/20 is the smallest inclusive summary.

Question 4: Answer

This question begins with the following routes that need to be summarized:

172.16.200.0/23
172.16.204.0/23
172.16.208.0/23

Note that the subnets are not consecutive in this case, but the algorithm still works. The first two steps are as follows:

1) Y = 2, because there are 3 component routes, and 2^2 => 3
2) Start with a prefix length of 23 – 2 = 21

From there, Table D-96 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the right answer is found.

Table D-96 *Question 4 Answer: Inclusive Summary of 3 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
21	172.16.200.0/21 yields a subnet of 172.16.200.0/21	172.16.208.0/21 yields a subnet of 172.16.208.0/21
20	172.16.200.0/20 yields a subnet of 172.16.192.0/20	172.16.208.0/20 yields a subnet of 172.16.208.0/20
19	172.16.200.0/19 yields a subnet of 172.16.192.0/19	172.16.208.0/19 yields a subnet of 172.16.192.0/19

After three passes through Steps 3 and 4, the results are equal, implying that 172.16.192.0/19 is the smallest inclusive summary.

Question 5: Answer

This question begins with the following routes that need to be summarized:

172.16.200.0/23
172.16.202.0/23
172.16.204.0/23
172.16.206.0/23
172.16.208.0/23

The first two steps are as follows:

1) Y = 3, because there are 5 component routes, and 2^3 => 5
2) Start with a prefix length of 23 – 3 = 20

From there, Table D-97 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the right answer is found.

Table D-97 *Question 5 Answer: Inclusive Summary of 5 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
20	172.16.200.0/20 yields a subnet of 172.16.192.0/20	172.16.208.0/20 yields a subnet of 172.16.208.0/20
19	172.16.200.0/19 yields a subnet of 172.16.192.0/19	172.16.208.0/19 yields a subnet of 172.16.192.0/19

After two passes through Steps 3 and 4, the results are equal, implying that 172.16.192.0/19 is the smallest inclusive summary.

Question 6: Answer

This question begins with the following routes that need to be summarized:

172.16.112.0/22
172.16.116.0/22
172.16.120.0/22
172.16.124.0/22

The first two steps are as follows:

1) Y = 2, because there are 4 component routes, and 2^2 => 4
2) Start with a prefix length of 22 – 2 = 20

From there, Table D-98 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the right answer is found.

Table D-98 *Question 6 Answer: Inclusive Summary of 4 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
20	172.16.112.0/20 yields a subnet of 172.16.112.0/20	172.16.124.0/20 yields a subnet of 172.16.112.0/20

Question 7: Answer

This question begins with the following routes that need to be summarized:

192.168.1.16/29
192.168.1.24/29
192.168.1.32/29

The first two steps are as follows:

1) Y = 2, because there are 3 component routes, and 2^2 => 3
2) Start with a prefix length of 29 – 2 = 27

From there, Table D-99 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the right answer is found.

Table D-99 *Question 7 Answer: Inclusive Summary of 3 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
27	192.168.1.16/27 yields a subnet of 192.168.1.0/27	192.168.1.32/27 yields a subnet of 192.168.1.32/27
26	192.168.1.16/26 yields a subnet of 192.168.1.0/26	192.168.1.32/26 yields a subnet of 192.168.1.0/26

Question 8: Answer

This question begins with the following routes that need to be summarized:

192.168.1.16/28
192.168.1.32/28

The first two steps are as follows:

1) Y = 1, because there are 2 component routes, and 2^1 => 2
2) Start with a prefix length of 28 – 1 = 27

From there, Table D-100 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the right answer is found.

Table D-100 *Question 8 Answer: Inclusive Summary of 2 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
27	192.168.1.16/27 yields a subnet of 192.168.1.0/27	192.168.1.32/27 yields a subnet of 192.168.1.32/27
26	192.168.1.16/26 yields a subnet of 192.168.1.0/26	192.168.1.32/26 yields a subnet of 192.168.1.0/26

Question 9: Answer

This question begins with the following routes that need to be summarized:

10.1.80.0/25
10.1.81.0/25
10.1.81.128/25

The first two steps are as follows:

1) Y = 2, because there are 3 component routes, and 2^2 => 3
2) Start with a prefix length of 25 – 2 = 23

From there, Table D-101 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the right answer is found.

Table D-101 *Question 9 Answer: Inclusive Summary of 3 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
23	10.1.80.0/23 yields a subnet of 10.1.80.0/23	10.1.81.128/23 yields a subnet of 10.1.80.0/23

Question 10: Answer

This question begins with the following routes that need to be summarized:

10.1.80.0/26
10.1.81.0/26
10.1.81.128/26

The first two steps are as follows:

1) Y = 2, because there are 3 component routes, and 2^2 => 3
2) Start with a prefix length of 26 – 2 = 24

From there, Table D-102 shows the iterations through Steps 3 and 4, using progressively shorter prefix lengths, until the right answer is found.

Table D-102 *Question 10 Answer: Inclusive Summary of 3 Routes*

Prefix Length	Step 3 (Lowest Component Subnet)	Step 4 (Highest Component Subnet)
24	10.1.80.0/24 yields a subnet of 10.1.80.0/24	10.1.81.128/24 yields a subnet of 10.1.81.0/24
23	10.1.80.0/23 yields a subnet of 10.1.80.0/23	10.1.81.128/23 yields a subnet of 10.1.80.0/23

Discovering the Smallest Exclusive Summary Routes: 5 Questions

The last section of this appendix provides practice problems and answers for finding exclusive summaries. Per Chapter 4's conventions, an exclusive summary may include multiple prefixes/subnets, but it may only include address ranges inside the original component prefixes/subnets.

For the following five lists of subnets, discover the set of exclusive summary routes:

1. 10.20.7.0/24, 10.20.4.0/24, 10.20.5.0/24, 10.20.6.0/24
2. 10.20.3.0/24, 10.20.4.0/24, 10.20.5.0/24, 10.20.6.0/24, 10.20.7.0/24, 10.20.8.0/24
3. 172.16.200.0/23, 172.16.204.0/23, 172.16.208.0/23, 172.16.202.0/23, 172.16.206.0/23
4. 172.16.120.0/22, 172.16.112.0/22, 172.16.124.0/22, 172.16.116.0/22
5. 192.168.1.16/29, 192.168.1.32/29, 192.168.1.24/29

The following steps are a repeat of the decimal algorithm for finding exclusive summaries found in Chapter 4. Remember, the process assumes that all the component subnets have the same mask/prefix length.

Step 1 Find the best *inclusive* summary route; call it a *candidate exclusive* summary route.

Step 2 Determine if the candidate summary includes any address ranges it should not. To do so, compare the summary's implied address range with the implied address ranges of the component subnets.

Step 3 If the candidate summary only includes addresses in the ranges implied by the component subnets, the candidate summary is part of the best exclusive summarization of the original component subnets.

Step 4 If instead the candidate summary includes some addresses matching the candidate summary routes, and some addresses that do not match, split the current candidate summary in half, into two new candidate summary routes, each with a prefix 1 *longer* than before.

Step 5 If the candidate summary only includes addresses outside the ranges implied by the component subnets, the candidate summary is not part of the best exclusive summarization, and it should not be split further.

Step 6 Repeat Steps 2–4 for each of the two possible candidate summary routes created at Step 4.

Question 1: Answer

This question begins with the following routes that need to be summarized:

10.20.4.0/24, range 10.20.4.0–10.20.4.255
10.20.5.0/24, range 10.20.5.0–10.20.5.255
10.20.6.0/24, range 10.20.6.0–10.20.6.255
10.20.7.0/24, range 10.20.7.0–10.20.7.255

The inclusive summary for these routes is

10.20.4.0/22

Table D-103 shows what turns out to be a single pass through the algorithm, because the inclusive summary and exclusive summary are the same for this problem.

Table D-103 *Question 1 Answer: Exclusive Summary of 4 Routes*

Split	Candidate Exclusive Summary	Range of Addresses	Analysis
Inclusive summary	10.20.4.0/22	10.20.4.0–10.20.7.255	Part of exclusive summary

Comparing the range of IP addresses in the problem statement with the range of addresses implied by the original inclusive summary, you can see that it is the exact same set of addresses. As a result, 10.20.4.0/22 is part of the exclusive summary—in fact, no other summary routes are required.

Question 2: Answer

This question begins with the following routes that need to be summarized:

10.20.3.0/24, range 10.20.3.0–10.20.3.255
10.20.4.0/24, range 10.20.4.0–10.20.4.255
10.20.5.0/24, range 10.20.5.0–10.20.5.255
10.20.6.0/24, range 10.20.6.0–10.20.6.255
10.20.7.0/24, range 10.20.7.0–10.20.7.255
10.20.8.0/24, range 10.20.8.0–10.20.8.255

The inclusive summary for these routes is

10.20.0.0/20

Table D-104 begins by showing three passes through the algorithm. These three passes do not determine all the exclusive summary routes in the answer; Tables D-105 and D-106 complete the answer.

Before examining Table D-104, first consider the overall flow of the repeated iterations through the table. Think of the original inclusive summary route as one large group of addresses. If it is not also the exclusive summary, you iterate through the algorithm again, halving the original inclusive summary. If that does not produce an answer, you halve each of the halves for the next iteration through the algorithm. So, you can think of the second splitting of the candidate summaries as breaking them into quarters. Another pass would break the original inclusive summary into eighths, and so on. The table's first column denotes what each row means based on whether it is for the original inclusive summary, the first split (into halves), the second split (into quarters), and so on.

Table D-104 *Question 2 Answer: Inclusive Summary of 6 Routes, Part 1*

Split	Candidate Exclusive Summary	Range	Analysis
Inclusive summary	10.20.0.0/20	10.20.0.0–10.20.15.255	Includes too many addresses
1st split, lower half	10.20.0.0/21	10.20.0.0–10.20.7.255	Includes 10.20.0.0–10.20.2.255, which should not be included
1st split, higher half	10.20.8.0/21	10.20.8.0–10.20.15.255	Includes 10.20.9.0–10.20.15.255, which should not be included
2nd split, lowest quarter	10.20.0.0/22	10.20.0.0–10.20.3.255	Includes 10.20.0.0–10.20.2.255, which should not be included
2nd split, 2nd quarter	10.20.4.0/22	10.20.4.0–10.20.7.255	Includes only 10.20.4.0–10.20.7.255; it is **part of exclusive summary**
2nd split, 3rd quarter	10.20.8.0/22	10.20.8.0–10.20.11.255	Includes 10.20.9.0–10.20.11.255, which should not be included
2nd split, highest quarter	10.20.12.0/22	10.20.12.0–10.20.15.255	Includes 10.20.12.0–10.20.15.255, totally outside the range—**don't split again**

The last four rows of the table show the results of the second split (per Step 4 in the algorithm). Two of these four candidate exclusive summaries need to be split again (10.20.0.0/22 and 10.20.8.0/22) because they contain some addresses within the original ranges, but some outside the range. One summary (10.20.4.0/22) holds only addresses inside the original ranges, so that route is one of the routes comprising the exclusive summary. Finally, one candidate route (10.20.12.0/22) contains only addresses outside the original range; as a result, you can stop splitting that range when looking for the exclusive summaries.

Tables D-105 and D-106 complete the official algorithm, but through some basic inspection, you might be able to (rightfully) guess that no additional summary routes will be found. Consider the original routes, and whether the process has found a summary route to include the addresses yet:

10.20.3.0/24—still looking for summary
10.20.4.0/24—found summary
10.20.5.0/24—found summary
10.20.6.0/24—found summary
10.20.7.0/24—found summary
10.20.8.0/24—still looking for summary

Thinking about the problem from this point forward, the remaining component subnets—10.20.3.0/24 and 10.20.8.0/24—are separated by the previously discovered 10.20.4.0/22 summary. There is only one original route on each side of that summary. So, there is no possibility of summarizing those two individual routes.

The algorithm will reach that same conclusion, as shown in the next two tables. The third split is in Table D-105 (Table D-104 showed up through the second split), and the fourth split is in Table D-106. Keep in mind that, per Table D-104, only two prefixes need splitting for the next step in the process—10.20.0.0/22 and 10.20.8.0/22. The "Split" column in the table lists the halves of these two prefixes.

Table D-105 *Question 2 Answer, 3rd Split*

Split	Candidate Exclusive Summary	Range	Analysis
Lower half of 10.20.0.0/22	10.20.0.0/23	10.20.0.0–10.20.1.255	Holds none of the original addresses—**don't split again**
Higher half of 10.20.0.0/22	10.20.2.0/23	10.20.2.0–10.20.3.255	Includes too many addresses—**split again**
Lower half of 10.20.8.0/22	10.20.8.0/23	10.20.8.0–10.20.9.255	Includes too many addresses—**split again**
Higher half of 10.20.8.0/22	10.20.10.0/23	10.20.10.0–10.20.11.255	Holds none of the original addresses—**don't split again**

(Note: Per Table D-105, only 10.20.2.0/23 and 10.20.8.0/23 need splitting; their halves are noted in the first column.)

Table D-106 *Question 2 Answer: 4th Split*

Split	Candidate Exclusive Summary	Range	Analysis
Lower half of 10.20.2.0/23	10.20.2.0/24	10.20.2.0–10.20.2.255	Holds none of the original addresses—don't split again
Higher half of 10.20.2.0/23	10.20.3.0/24	10.20.3.0–10.20.3.255	**Part of exclusive summary**
Lower half of 10.20.8.0/23	10.20.8.0/24	10.20.8.0–10.20.8.255	**Part of exclusive summary**
Higher half of 10.20.8.0/23	10.20.9.0/23	10.20.9.0–10.20.9.255	Holds none of the original addresses—don't split again

The other two components of the set of exclusive summary routes are finally found in Table D-106. As a result, looking at all three tables, the answer for this question is as follows:

10.20.3.0/24
10.20.4.0/22
10.20.8.0/24

Question 3: Answer

This question begins with the following routes that need to be summarized:

172.16.200.0/23, range 172.16.200.0–172.16.201.255
172.16.202.0/23, range 172.16.202.0–172.16.203.255
172.16.204.0/23, range 172.16.204.0–172.16.205.255
172.16.206.0/23, range 172.16.206.0–172.16.207.255
172.16.208.0/23, range 172.16.208.0–172.16.209.255

The inclusive summary for these routes is

172.16.192.0/19

Table D-107 begins by showing three passes through the algorithm. These three passes do not determine all the summary routes in the answer.

Table D-107 *Question 3 Answer: Inclusive Summary of 5 Routes*

Split	Candidate Exclusive Summary	Range	Analysis
Inclusive summary	172.16.192.0/19	172.16.192.0–172.16.223.255	Includes too many addresses
1st split, lower half	172.16.192.0/20	172.16.192.0–172.16.207.255	Includes 172.16.192.0–172.16.199.255, which should not be included
1st split, higher half	172.16.208.0/20	172.16.208.0–172.16.223.255	Includes 172.16.210.0–172.16.223.255, which should not be included
2nd split, lowest quarter	172.16.192.0/21	172.16.192.0–172.16.199.255	Includes only address totally outside the range—**don't split again**
2nd split, 2nd quarter	172.16.200.0/21	172.16.200.0–172.16.207.255	Includes only address in the range—**it's part of exclusive summary**

Table D-107 *Question 3 Answer: Inclusive Summary of 5 Routes (Continued)*

Split	Candidate Exclusive Summary	Range	Analysis
2nd split, 3rd quarter	172.16.208.0/21	172.16.208.0–172.16.215.255	Includes some addresses that should not be included
2nd split, highest quarter	172.16.216.0/21	172.16.216.0–172.16.223.255	Includes only address totally outside the range—**don't split again**

The last four rows of the table show the results of the second split (per Step 4 in the algorithm). Two of these four candidate exclusive summaries (172.16.192.0/21 and 172.16.216.0/21) only contain addresses outside the range that needs to be summarized, so these do not need to be split further. 172.16.200.0/21 is part of the exclusive summary, so it does not need to be split again. Only 172.16.208.0/21 needs further splitting at this point.

Under closer examination, at this point in the process, no further work is actually needed. Only one original component subnet has not had its address range summarized. For reference, the following list describes which ranges are part of the one exclusive summary route that has already been uncovered (172.16.200.0/21), and those that are not inside that summary route:

172.16.200.0/24—part of summary 172.16.200.0/21
172.16.202.0/24—part of summary 172.16.200.0/21
172.16.204.0/24—part of summary 172.16.200.0/21
172.16.206.0/24—part of summary 172.16.200.0/21
172.16.208.0/24—still looking for summary

Because only one component subnet still needs to be summarized, there is no possibility that a larger exclusive summary route will be found, because there are no other component subnets to combine with 172.16.208.0/24. As a result, the final answer for this problem (the exclusive summary routes for the component subnets) is as follows:

172.16.200.0/21
172.16.208.0/24

Question 4: Answer

This question begins with the following routes that need to be summarized:

172.16.112.0/22, range 172.16.112.0–172.16.115.255
172.16.116.0/22, range 172.16.116.0–172.16.119.255
172.16.120.0/22, range 172.16.120.0–172.16.123.255
172.16.124.0/22, range 172.16.124.0–172.16.127.255

The inclusive summary for these routes is

172.16.112.0/20, range 172.16.112.0–172.16.127.255

By simply inspecting the inclusive summary, you can see that it exactly matches the collective ranges of IP addresses in the four component subnets. So, the exclusive summary for these four subnets is also 172.16.112.0/20.

Question 5: Answer

This question begins with the following routes that need to be summarized:

192.168.1.16/29, range 192.168.1.16–192.168.1.23
192.168.1.24/29, range 192.168.1.24–192.168.1.31
192.168.1.32/29, range 192.168.1.32–192.168.1.39

The inclusive summary for these routes is

192.168.1.0/26

Table D-108 begins by showing three passes through the algorithm. These three passes do not determine all the summary routes in the answer.

Table D-108 *Question 2 Answer: Inclusive Summary of 3 Routes*

Split	Candidate Exclusive Summary	Range	Analysis
Inclusive summary	192.168.1.0/26	192.168.1.0–192.168.1.63	Includes too many addresses
1st split, lower half	192.168.1.0/27	192.168.1.0–192.168.1.31	Includes too many addresses—split again
1st split, higher half	192.168.1.32/27	192.168.1.32–192.168.1.63	Includes too many addresses—split again
2nd split, lowest quarter	192.168.1.0/28	192.168.1.0–192.168.1.15	Includes only address totally outside the range—**don't split again**
2nd split, 2nd quarter	192.168.1.16/28	192.168.1.16–192.168.1.31	Includes only address in the range—**it's part of exclusive summary**
2nd split, 3rd quarter	192.168.1.32/28	192.168.1.32–192.168.1.47	Includes some addresses that should not be included
2nd split, highest quarter	192.168.1.48/28	192.168.1.48–192.168.1.63	Includes only address totally outside the range—**don't split again**

The last four rows of the table show the results of the second split (per Step 4 in the algorithm). Two of these four candidate exclusive summaries (192.168.1.0/28 and 192.168.1.48/28) only contain addresses outside the range that needs to be summarized, so these do not need to be split further. 192.168.1.16/28 is part of the exclusive summary, so it does not need to be split again. Only 192.168.32.0/28 needs further splitting at this point.

Under closer examination, at this point in the process, no further work is actually needed. Only one original component subnet has not had its address range summarized. For reference, the following list describes which ranges are part of the one exclusive summary route that has already been uncovered (192.168.1.16/28), and those that are not inside that summary route:

192.168.1.16/29—part of summary 192.168.1.16/28
192.168.1.24/29—part of summary 192.168.1.16/28
192.168.1.16/29—still looking for summary

Because only one component subnet still needs to be summarized, there is no possibility that a larger exclusive summary route will be found. As a result, the final answer for this problem (the exclusive summary routes for the component subnets) is as follows:

192.168.1.16/28
192.168.1.32/29